Attorney Docket No. IGT1P071/P-677

PATENT APPLICATION

DIGITAL IDENTIFICATION OF UNIQUE GAME CHARACTERISTICS

Inventor(s): Richard E. Rowe
235 Bluewater Court
Reno, Nevada 89509
U.S. Citizen

Assignee: International Game Technology

BEYER WEAVER & THOMAS, LLP
P.O. Box 778
Berkeley, California 94704-0778
(510) 843-6200

PATENT

Attorney Docket No. IGT1P071/P-677

DIGITAL IDENTIFICATION OF UNIQUE GAME CHARACTERISTICS

RELATED APPLICATION DATA

The present application relates generally to U.S. Patent Application No. 09/746,944 for GAMING TERMINAL DATA REPOSITORY AND INFORMATION DISTRIBUTION SYSTEM filed on December 21, 2000, the entire disclosure of which is incorporated herein by reference for all purposes.

BACKGROUND OF THE INVENTION

This invention relates to gaming network including gaming machines such as video slot machines and video poker machines. More specifically, the present invention provides methods and apparatus for verifying the authenticity of distributed gaming applications having a plurality of associated gaming application objects.

Typically, a master gaming controller in a gaming machine controls various combinations of devices that allow a player to play a game on the gaming machine and encourage game play on the gaming machine. For example, a game played on a gaming machine usually requires a player to input money or indicia of credit into the gaming machine, indicate a wager amount, and initiate a game play. These steps require the gaming machine to control input devices, including bill validators and coin acceptors, to accept money into the gaming machine and recognize user inputs from devices, including touch screens and button pads, to determine the wager amount and initiate game play. After game play has been initiated, the gaming machine determines a game outcome, presents the game

outcome to the player and may dispense an award of some type depending on the outcome of the game.

As technology in the gaming industry progresses, the traditional mechanically driven reel slot machines are being replaced with electronic counterparts having CRT, LCD video displays or the like and gaming machines such as video slot machines and video poker machines are becoming increasingly popular. Part of the reason for their increased popularity is the nearly endless variety of games that can be implemented on gaming machines utilizing advanced electronic technology. In some cases, newer gaming machines are utilizing computing architectures developed for personal computers. These video/electronic gaming advancements enable the operation of more complex games, which would not otherwise be possible on mechanical-driven gaming machines and allow the capabilities of the gaming machine to evolve with advances in the personal computing industry.

When implementing the gaming features described above on a gaming machine using architectures utilized in the personal computer industry, a number of requirements unique to the gaming industry must be considered. One such requirement is the regulation of gaming software. Typically, within a geographic area allowing gaming, i.e. a gaming jurisdiction, a regulatory body is charged with regulating the games played in the gaming jurisdiction to ensure fairness and prevent cheating. In most gaming jurisdictions there are stringent regulatory restrictions for gaming machines requiring a time consuming approval process of new gaming software and any software modifications to gaming software used on a gaming machine. A regulatory scheme also typically includes field verification of deployed gaming applications to ensure that a deployed game corresponds to the certified version of the game.

In the past, to implement the play of a game on a gaming machine, a monolithic software architecture has been used. In a monolithic software architecture, a single gaming

software executable is developed. The single executable is typically burnt into an EPROM and then submitted to various gaming jurisdictions for approval. After the gaming application is approved, a unique checksum is determined for the gaming application stored in the EPROM for the purpose of uniquely identifying the approved version of the gaming application.

A disadvantage of a monolithic programming architecture is that a single executable that works for many different applications can be quite large. For instance, gaming rules may vary from jurisdiction to jurisdiction. Thus, either a single custom executable can be developed for each jurisdiction or one large executable with additional logic can be developed that is valid in many jurisdictions. The customization process may be time consuming and inefficient. For instance, upgrading the gaming software may require developing new executables for each jurisdiction, submitting the executables for reapproval, and then replacing or reprogramming EPROMs in each gaming machine.

By contrast, software architectures for use by personal computers have moved toward an object oriented approach where different software objects may be dynamically linked together prior to or during execution to create many different combinations of executables that perform different functions. Thus, for example, to account for differences in gaming rules between different gaming jurisdictions, gaming software objects appropriate to a particular gaming jurisdiction may be linked at run-time which is simpler than creating a single different executable for each jurisdiction. Also, object oriented software architectures simplify the process of upgrading software since a software object, which usually represent only a small portion of the software, may be upgraded rather than the entire software.

However, object oriented software architectures are not compatible with the traditional gaming industry approach of storing static executables in EPROMs. As a result, the gaming software regulation process described above using EPROM checksums is largely

inapplicable to the future of the gaming industry. That is, regulators in the gaming industry employ a device know as a "Cobatron" which they use to generate the checksum for an EPROM which is actually physically removed from a randomly selected gaming machine. Because the checksum uniquely identifies the gaming application stored in the EPROM, the regulator can verify that the application in the selected gaming machine is the application certified by the gaming commission, thus ensuring the integrity of the gaming machine.

Obviously, this approach is not applicable to a distributed gaming environment in which a gaming application may correspond to a collection of objects, some of which are downloaded to the gaming machine from a remote server or the Internet. Therefore, as gaming technology moves toward more distributed architectures, there is a need for techniques by which gaming applications corresponding to multiple objects may be uniquely identified for regulatory and other purposes.

SUMMARY OF THE INVENTION

According to various embodiments of the present invention, methods and apparatus are provided for generating a gaming application signature which uniquely represents a gaming application having a plurality of gaming application objects associated therewith. A subset of the plurality of gaming application objects are retrieved. An object signature is generated for each of the retrieved gaming application objects. The object signatures are combined to generate the gaming application signature.

According to a specific embodiment, the gaming application signature comprises an original signature which is then stored for authentication of subsequently generated signatures corresponding to deployed gaming applications. According to another specific embodiment, the gaming application signature corresponds to a deployed gaming application which is compared to a previously stored original signature to authenticate the gaming application.

A further understanding of the nature and advantages of the present invention may be realized by reference to the remaining portions of the specification and the drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1A is block diagram of a gaming machine.

Figs. 1B and 1C are block diagrams of gaming machines connected to remote storage devices.

Fig. 2 is a perspective drawing of a gaming machine having a top box and other devices.

Fig. 3 is a block diagram of a gaming process file structure.

Fig. 4 is a flow chart illustrating an exemplary process by which an original gaming application signature may be generated.

Fig. 5 is a flow chart illustrating an exemplary process by which a previously generated gaming application signature is used to authenticate a corresponding gaming application deployed in the field.

DETAILED DESCRIPTION OF SPECIFIC EMBODIMENTS

According to the present invention, methods and apparatus are provided by which a gaming application corresponding to a plurality of software objects may be uniquely identified. According to a specific embodiment, unique signatures for at least some of the objects corresponding to a gaming application are combined to create a unique ID by which a regulator may verify that the certified version of the gaming application is the one which has been deployed. According to various specific embodiments, the unique signatures for the individual objects may include conventionally generated digital signatures for audio, video, graphic, and other objects. These signatures may then be combined in a wide variety of ways to generate the unique ID.

The digital signatures by which each object associated with a gaming application is identified may be generated using any of a wide variety of conventional or proprietary techniques. For example, a digital signature for an audio file may be generated using audio sampling techniques such as those described in U.S. Patent No. 6,243,480 (Zhao et al.), the entire disclosure of which is incorporated herein by reference for all purposes. Similarly, a digital signature for a video file may be generated using techniques described in U.S. Patent No. 6,229,924 (Rhoads et al.), the entire disclosure of which is incorporated herein by reference for all purposes. Digital water marks such as those described in U.S. Patent No. 6,163,842 (Barton), the entire disclosure of which is incorporated herein by reference for all purposes, may be embedded in the various objects associated with a gaming application. These water marks could then be extracted, combined, and compared to a similarly created signature associated with the authorized version of the gaming application. Alternatively, proprietary algorithms may be created which generate a unique data structure corresponding to the object in question. Even conventional checksum algorithms may be applied to the

objects to generate individual checksums which may then be combined to create the unique ID.

The method by which these individual object signatures are combined can similarly employ any of a wide variety of mechanisms. Such mechanisms may include, for example, a simple logic function, e.g., an exclusive-OR, more complex logic functions, any of a variety of standard checksum or hashing function algorithms, or any technique which generates a unique data structure for the combination of signatures.

Fig. 1A is block diagram of a gaming machine 102 for use with one embodiment of the present invention. A master gaming controller 101 is used to present one or more games on the gaming machine 102. The master gaming controller 101 executes a number of gaming software programs to operate gaming devices 112 such as coin hoppers, bill validators, coin acceptors, speakers, printers, lights, displays (e.g. 110) and input mechanisms. One or more displays, such as 110, may be used on the gaming machine. The one or more displays may be mechanical displays (e.g., slot reels), video displays or combinations thereof. The master gaming controller 101 may execute gaming software enabling complex graphical renderings to be presented on one or more displays that may be used as part of a game outcome presentation on the gaming machine 102. The master gaming controller 101 may also execute gaming software enabling communications with gaming devices located outside of the gaming machine 102, such as player tracking servers and progressive game servers. In some embodiments, communications with devices located outside of the gaming machine may be performed using the main communication board 108 and network connection 125.

In conjunction with various embodiments of the present invention, gaming software executed on the gaming machine 102 by the master gaming controller 101 may be regularly verified by comparing software stored in RAM 106 for execution on the gaming machine

102 with certified copies of the software stored on the gaming machine, accessible to the gaming machine via a remote communication connection or combinations thereof. Two gaming software units are used to implement this method: 1) a code comparator and 2) a code authenticator. The code comparator compares at least some portion of the gaming software scheduled for execution on the gaming machine at a particular time with authenticated gaming software stored in a file storage media accessible to the gaming machine 102. The file storage media may comprise one or more file storage devices located on the gaming machine 102, on other gaming machines, on remote servers or combinations thereof. During operation of the gaming machine, the code comparator periodically checks the gaming software programs being executed by the master gaming controller 101 as the gaming software programs executed by the master gaming controller 101 may vary with time.

The code authenticator locates on the file storage media an authentic copy of the gaming software being checked by the code comparator. During the boot process for the gaming machine 102 the code authenticator may be loaded from an EPROM such as 104. The master gaming controller 101 executes various gaming software programs using one or more processors such as CPU 103. During execution, a software program may be temporarily loaded into the RAM 106. Depending on the current operational state of the gaming machine, the number and types of software programs loaded in the RAM 106 may vary with time. For instance, when a game is presented, particular software programs used to present a complex graphical presentation may be loaded into RAM 106. However, when the gaming machine 102 is idle, these graphical software programs may not be loaded into the RAM.

The code authenticator searches a file system available to the gaming machine for certified/authentic copies of gaming software programs currently being executed by the

gaming machine. The file system may be distributed across one or more file storage devices. The certified/authentic copies of gaming software programs may be certified after a regulatory approval process as described above. The certified/authentic copies of gaming software programs may be stored in a "static" mode (e.g. read-only) on one or more file storage devices located on the gaming machine 102 such as file storage device 114 or EPROM 104. The file storage devices may be a hard-drive, CD-ROM, CD-DVD, static RAM, flash memory, EPROMs or combinations thereof.

The file system used by the code authenticator may be distributed between file storage devices located on the gaming machine or on remote file storage devices. Figs. 1B and 1C are block diagrams of gaming machines connected to remote storage devices. In Fig. 1B, gaming machine 102 is connected to two remote file storage devices 116 and 118. The code authenticator may search the two remote file storage devices 116 and 118 as well as local file storage device 114 for gaming software programs that correspond to gaming software programs currently scheduled for execution by the master gaming controller 101. Using a resource sharing system, a number of gaming software programs may be simultaneously scheduled for execution on the gaming machine at any one time. The resource sharing system, usually embedded in the operating system, develops a sequence order for executing the combination of gaming software programs. When the code authenticator returns a file name and file location (e.g. one of the file storage devices), the code comparator may compare portions of the software program being executed on the gaming machine with a corresponding software program stored one of the file storage devices. The gaming software programs identified by the code authenticator may be in an executable "object" format that includes programming instructions substantially identical to the format of the programming instructions executing on the gaming machine.

According to a specific embodiment of the present invention, a majority of gaming software programs used on the gaming machine are stored on a remote device such as a game server. In Fig. 1C, three gaming machines, 120, 121 and 122 are connected to a game server 124. In this example, the gaming machines 120, 121 and 122 do not include a local file storage device such as a hard drive and gaming executables are downloaded from the game server 124. On each of the gaming machines 120, 121 and 122, the code comparator may compare software being executed by the gaming machine with certified/authentic code stored on the game server 124.

Turning to Fig. 2, a video gaming machine 200 for use with specific embodiments of the present invention is shown. Machine 200 includes a main cabinet 204, which generally surrounds the machine interior (not shown) and is viewable by users. The main cabinet includes a main door 208 on the front of the machine, which opens to provide access to the interior of the machine. Attached to the main door are player-input switches or buttons 232, a coin acceptor 228, and a bill validator 230, a coin tray 238, and a belly glass 240. Viewable through the main door is a video display monitor 234 and an information panel 236. The display monitor 234 will typically be a cathode ray tube, high resolution flat-panel LCD, or other conventional electronically controlled video monitor. The information panel 236 may be a back-lit, silk screened glass panel with lettering to indicate general game information including, for example, a game denomination (e.g. $.25 or $1). The bill validator 230, player-input switches 232, video display monitor 234, and information panel are devices used to play a game on the game machine 200. The devices are controlled by circuitry (See Fig. 1A) housed inside the main cabinet 204 of the machine 200. Many possible games, including mechanical slot games, video slot games, video poker, video black jack, video pachinko, video bingo, video card games, lottery, and other games of chance may be provided with gaming machines of this invention.

The gaming machine 200 includes a top box 206, which sits on top of the main cabinet 204. The top box 206 houses a number of devices, which may be used to add features to a game being played on the gaming machine 200, including but not limited to: a) speakers 210, 212, 214, a ticket printer 218 which prints bar-coded tickets 220, b) a key pad 222 for entering player tracking information such as an identification code, c) a florescent display 216 for displaying player tracking information, d) a card reader 224 for entering a magnetic striped card containing player tracking information or other input devices for entering player tracking information, e) a speaker/microphone for voice commands and voice recognition, f) biometric input devices such as finger printer for identifying a player, g) a video display screen 244 for displaying various types of video content such as player tracking information, machine status, bonus games and primary games and h) a lighted candle that may be used for signaling purposes such as to get the attention of various casino personnel. In some embodiments, some of these gaming devices may also be incorporated into the main cabinet of the gaming machine 200. The ticket printer 218 may be used to print tickets for a cashless ticketing system. Further, the top box 206 may house different or additional devices than those shown. For example, the top box may contain a bonus wheel or a back-lit silk screened panel which may be used to add bonus features to the game being played on the gaming machine. As another example, the top box may contain a display for a progressive jackpot offered on the gaming machine. During a game, these devices are controlled and powered, in part, by circuitry (See Fig. 1A) housed within the main cabinet 204 of the machine 200.

It should be noted that that gaming machine 200 is but one example from a wide range of gaming machine designs with which the present invention may be practiced. For example, not all suitable gaming machines have top boxes or player tracking features. Further, some gaming machines have two or more game displays – mechanical and/or video.

And, some gaming machines are designed for bar tables and have displays that face upwards. As another example, a game may be generated on a host computer and may be displayed on a remote terminal or a remote computer. The remote computer may be connected to the host computer via a network of some type such as the Internet. Those of skill in the art will understand that the present invention, as described below, can be deployed on most any gaming machine now available or hereafter developed.

Returning to the example of Fig. 2, when a user wishes to play the gaming machine 200, he or she inserts cash through the coin acceptor 228 or bill validator 230. Additionally, the bill validator may accept a printed ticket voucher which may be accepted by the bill validator 230 as an indicia of credit when a cashless ticketing system is used. At the start of the game, the player may enter playing tracking information using the card reader 224, the keypad 222, and the florescent display 216. Further, other game preferences of the player playing the game may be read from a card inserted into the card reader. During the game, the player views game information using the video display 234. Other game and prize information may also be displayed in the video display screen 244 located in the top box 206.

During the course of a game, a player may be required to make a number of decisions, which affect the outcome of the game. For example, a player may vary his or her wager on a particular game, select a prize for a particular game selected from a prize server, or make game decisions which affect the outcome of a particular game. The player may make these choices using the player-input switches 232, the video display screen 234 or using some other device which enables a player to input information into the gaming machine. In some embodiments, the player may be able to access various game services such as concierge services and entertainment content services using the video display screen 234 and one more input devices.

During certain game events, the gaming machine 200 may display visual and auditory effects that can be perceived by the player. These effects add to the excitement of a game, which makes a player more likely to continue playing. Auditory effects include various sounds that are projected by the speakers 210, 212, 214. Visual effects include flashing lights, strobing lights or other patterns displayed from lights on the gaming machine 200 or from lights behind the belly glass 240. After the player has completed a game, the player may receive game tokens from the coin tray 238 or the ticket 220 from the printer 218, which may be used for further games or to redeem a prize. Further, the player may receive a ticket 220 for food, merchandise, or games from the printer 218.

Fig. 3 is a block diagram of a gaming process file structure 300. As a player utilizes a gaming machine in the manner described above, many different software programs may be executed by the gaming machine. As different gaming software programs are executed by the gaming machine, an operating system running on the gaming machine assign the programs memory location in RAM and then schedule and track the execution of each program as "processes."

In one example, every time a process is launched in the operating system, a special directory, such as 310, 315, 320, 325 and 330, is created under the directory "/proc" 305 (e.g. the process directory) in the operating system. The name of this directory is identical to the process ID number (PID) of the process. For instance, process directories corresponding to process ID numbers "1", "2", "4049", "1234" and "6296" are stored under the "/proc" 305 directory. The process directories listed under the "/proc" directory 305 may vary as a function of time as different processes are launched and other process are completed.

In one embodiment, under each PID directory, such as 310, 315, 320, 325 and 330, an address space (AS) file, titled "AS", may be stored. The AS files, such as 335, 340, 345,

350 and 355 may contains various information about its parent process. Items stored in this file may include, among other things, the command line name used to launch the program and it's location in RAM (e.g. 350) and the names and location in RAM of the shared objects (so) that the process uses (e.g. 352, 354 and 356). A shared object is a gaming software program that may be shared by a number of other gaming software programs.

The shared objects used by a process on the gaming machine may vary with time. Thus, the number of shared objects such as 352, 354 and 356 used by a process may vary with time. For instance, a process for a game presentation on a gaming machine may launch various graphical shared objects and audio shared objects during the presentation of a game on the gaming machine and various combinations of these shared objects may be used at various times in the game presentation. For example, a shared object for a bonus game presentation on the gaming machine may only be used when a bonus game is being presented on the gaming machine. Hence, a process for a bonus game presentation may be launched when a bonus game presentation is required and the process may terminate when the bonus game presentation is completed. When the game presentation process uses the bonus game presentation shared object, the launching and the termination of the bonus game presentation shared object may be reflected in the AS file for the game presentation process.

The code comparator may use the AS files to determine which game related processes are currently being executed on the gaming machine. The code comparator may also be a process designated in the "/proc" directory 305. Also, in the "/proc" directory there may exist one or more directories that are not representations of process Ids. These include, but are not limited to, SELF, boot 330, ipstats, mount, etc. When parsing the "/proc" directory, these directories are skipped as they do not represent game related code. Once a valid directory is found, e.g., "4049" 320, it is opened and the "AS" file in it may parsed.

During the gaming application certification process, the regulating body, e.g., the gaming commission, generates the gaming application signature against which the signatures of gaming applications operating in the field are subsequently measured. According to various embodiments of the present invention, a software program is employed to both generate the original gaming application signature and perform field comparisons between the original signature and signatures generated in the field.

Fig. 4 is a flowchart 400 illustrating an exemplary process by which an original gaming application signature may be generated. To generate the original signature, the representative of the gaming commission enters a game-specific command in a user interface (402) which causes some or all of the objects or software modules associated with the designated game to be retrieved (404). These objects may be identified and retrieved, for example, as described above with reference to Figs. 1B and 1C and the code authenticator. Such objects may include, for example, a core gaming application object, an audio object, a video object, a graphics object, a pay table object, etc. The user interface may provide access, for example, to a gaming application server on which the objects are stored. Alternatively, the user interface may provide access to multiple machines on a network amongst which the objects are distributed as described above with reference to Figs. 1B and 1C.

An object signature is generated for each of the retrieved objects according to one or more of a variety of techniques (406). For example, a standard checksum might be generated for the core gaming application object, a proprietary data structure for the pay table object, and conventional digital signatures for the audio and video objects as described above. It should be noted that, according to some embodiments, object signatures may only be generated for some subset of all of the objects associated with a gaming application. That

is, only some of the objects associated with the gaming application might be retrieved for this purpose.

The generated object signatures are then combined to form the gaming application signature (408). As mentioned above, this may be achieved using any of a wide variety of conventional and proprietary techniques. That is, the object signatures may be combined using a simple logic function such as an exclusive-OR, or a more complicated logic function using a combination of logic operations. Alternatively, a hashing function may be applied to the object signatures to generate the signature. As another alternative, a proprietary technique may be employed to combine the signatures into a unique data structure.

The gaming application signature is then stored for subsequent authentication of versions of the corresponding game in the field (410). That is, only gaming applications having the same signature will be considered in compliance with the gaming commission's certification scheme.

Fig. 5 is a flowchart 500 illustrating an exemplary process by which a previously generated gaming application signature is used to authenticate a corresponding gaming application deployed in the field. In response to a visit by a gaming industry regulator to a gaming establishment, access to the establishment's gaming application server (e.g., server 124 of Fig. 1C) is typically provided for the purpose of verifying compliance with the applicable gaming regulations (502). This access may be direct, i.e., via the network administrator's terminal, or indirect, e.g., via a particular gaming machine or other node on the establishment's gaming network.

Once access is provided, the regulator begins execution of the authentication process of the present invention by booting certified game authentication software and entering a game-specific command identifying the particular gaming application for which authentication is desired (504). In response to the command, some or all of the objects or

software modules associated with the designated game are retrieved (506). As mentioned above, a process similar to the code authenticator described above could facilitate the identification and retrieval of these objects which may include, for example, a core gaming application object, an audio object, a video object, a graphics object, a pay table object, etc. The objects may be retrieved, for example, from a gaming application server on which the objects are stored. Alternatively, the objects may be retrieved from multiple machines on the network amongst which the objects are distributed, e.g., the gaming application server and a specific gaming machine.

An object signature is generated for each of the retrieved objects according to one or more of a variety of techniques as described above (508). The generated object signatures are then combined to form a signature corresponding to the deployed gaming application (510). As mentioned above, this may be achieved using any of a wide variety of conventional and proprietary techniques.

The deployed gaming application signature is then compared to the previously stored original gaming application signature which corresponds to the version of the game certified by the gaming commission (512). If the comparison determines the signatures to be the same (514), the deployed gaming application is determined to be authentic (516) and an appropriate message is displayed (518). If, on the other hand, the comparison determines the signatures to be different (514), the deployed gaming application is determined to be an uncertified or invalid version of the game (520) and an appropriate message is displayed (522).

According to a specific embodiment, where the process of Fig. 5 determines the deployed gaming application to be invalid, the regulator may look at the individual object signatures and compare each to the corresponding component of the original certified

signature to determine which of the objects associated with the gaming application represent the noncompliance condition.

While the invention has been particularly shown and described with reference to specific embodiments thereof, it will be understood by those skilled in the art that changes in the form and details of the disclosed embodiments may be made without departing from the spirit or scope of the invention. For example, specific embodiments have been described herein with reference to particular gaming machine and gaming network architectures. However, it will be understood that the present invention may be implemented in a wide variety of gaming environments without departing from the scope of the invention. That is, a personal computer connected with a gaming server via the Internet is an appropriate environment in which the present application may be practiced. More generally, the present invention is equally applicable to a single property, e.g., a casino, gaming environment and an virtual gaming environment provided over a wide area network such as the Internet.

In addition, although various advantages, aspects, and objects of the present invention have been discussed herein with reference to various embodiments, it will be understood that the scope of the invention should not be limited by reference to such advantages, aspects, and objects. Rather, the scope of the invention should be determined with reference to the appended claims.